

Bridging Data Science Programming with Advanced Formal Coursework

Wesley A. Brashear
 HPRC†
 Texas A&M University
 College Station, TX US
 wbrashear@tamu.edu

Zhenhua He
 HPRC†
 Texas A&M University
 College Station, TX US
 happidence1@tamu.edu

Richard Lawrence
 HPRC†
 Texas A&M University
 College Station, TX US
 rarensu@tamu.edu

Dhruva K. Chakravorty
 HPRC†
 Texas A&M University
 College Station, TX US
 chakravorty@tamu.edu

Tatevik Sekhposyan
 Department of Economics
 Texas A&M University
 College Station, TX US
 tsekposyan@tamu.edu

Margaret L. Carpenter
 Department of Economics
 Texas A&M University
 College Station, TX US
 marge@tamu.edu

Honggao Liu
 HPRC†
 Texas A&M University
 College Station, TX US
 honggao@tamu.edu

ABSTRACT

In order to fulfill the needs of an evolving job market, formal academic programs are continuously expanding computational training in traditional discipline-specific courses. We developed an informal, twelve contact-hour course tailored for economics students entering a computationally rigorous graduate-level course to help mitigate disparities in computing knowledge between students and prepare them for more advanced instruction within the formal setting. The course was developed to teach the R programming language to students without assuming any prior knowledge or experience in programming or the R environment. In order to allow for ease of implementation across various training approaches, the course was modularized with each section containing distinct topics and learning objectives. These modules can be easily developed as independent lessons so that discipline-specific needs can be addressed through inclusion or exclusion of certain topics. This implementation used the R package ‘learnr’ to develop the course, which rendered a highly extensible and portable interactive Shiny document that can be deployed on any system on which RStudio is installed. The course is offered as a series of interactive sessions during which students are led through the Shiny notebook by an instructor. Owing to its structure, it can be offered as an asynchronous web-based set of tutorials as well.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Copyright ©JOCSE, a supported publication of the Shodor Education Foundation Inc.

© 2022 Journal of Computational Science Education
 DOI: <https://doi.org/10.22369/issn.2153-4136/13/2/1>

Keywords

R, Economics, Cybertraining, Shiny, Interactive computing

1. INTRODUCTION

The R programming language has continued to gain in popularity over the last several years [1]. It is widely used across numerous disciplines and is a common staple of scientific workflows. One of the reasons for this increase in popularity is the continuous development and publication of packages that extend the utility of the R programming language as a data science tool. While some of these packages are utilized by a wide range of users (e.g., ggplot2), others are restricted to specific disciplines or areas of research and require extensive education within those disciplines for users to understand and utilize the increased functionality. This creates a gap for potential users who have the requisite knowledge and desire to utilize discipline-specific packages but lack the foundational knowledge necessary for working in the R programming environment. Responding to the need to bridge the gap for students who are both new to R programming and require the use of these discipline-specific R packages, we created a twelve-hour non-credit-bearing workshop for students to prepare them for future formal courses that utilize R. Here, we focus on students enrolled in the Graduate program offered by the Department of Economics at Texas A&M University.

An early challenge faced when developing introductory-level curricula is keeping students engaged while learning the foundational concepts required for expansion into discipline-specific computational knowledge [12, 11]. The design of a course plays a large role in student engagement and should include a variety of teaching strategies; including lectures, guided hands-on examples, and independent exercises. Furthermore, teaching in incremental modules with independent learning objectives can foster a sense of productive progress for the students. Another

† High Performance Research Computing

means to increase engagement is to create guided or independent exercises that utilize data relevant to the discipline of the target audience (e.g., economic data) or to shared personal experiences (e.g., enrollment at a particular university).

Texas A&M High Performance Research Computing has abundant experience developing informal courses and pedagogical approaches tailored for specific user groups [3–10]. These courses cover a wide scope of topics including introductions to our HPC systems and Linux basics; programming in Python, R, Matlab, and Pearl; and bioinformatics. After the success of our recent twenty-contact-hour training sessions tailored for teaching Python to economics graduate students [9], we were invited to develop a similar course for the R programming language.

2. METHODS

The topics covered in the class (Table 1) were determined through conversations with faculty members of the Texas A&M Department of Economics, analysis of course syllabi, interviews with students, and surveys of online tutorials for R. Most of the students taking the course were doing so in preparation for an accredited course covering economic forecasting in the following semester. Others did so to demonstrate proficiency in the R programming language to prospective employers. The scope of the course was selected with the students' future coursework in mind. Taking a departure from our previous work that aimed to develop proficiency in Python programming (10), here the learning objectives and outcomes were geared toward ensuring that a student develops proficiency in relevant skills. As such, a guiding principle was to introduce topics with the view of ensuring high adoption among students so that they could use them during the semester to address domain-specific problems. For example, the students would be expected to be ready to learn `rmarkdown` and statistical methods during the workshop and later rely on this knowledge during their future coursework.

The course's structure builds on the HPRC short course program [4], and leverages the strengths of our Python training program for graduate students [9]. The course was developed for a virtual setting with scaffolding to ensure seamless transition to an in-person learning environment. The course consists of twelve contact hours taught over two days, broken into 50-minute increments. The format of the modules included a brief introduction to each topic (typically through Google Slides) followed by a hands-on, instructor-guided/led example and then independent exercises. Students were given short, one-question quizzes throughout each day to track participation. We used the package 'learnr' [1] to create a Shiny interactive document that allows students to create, edit, and run small code chunks in R. This format allows for instructional guidelines and helpful information to be displayed to students alongside windows for running code. This highly portable format can be deployed across any system with RStudio, including personal computers, on-campus research computing systems, and RStudio Cloud. Students were given the option of running the learnr document for this course on their own computer, in RStudio Cloud, or through Texas A&M's Virtual Open Access Lab (VOAL). VOAL allows Texas A&M students to connect remotely to a Windows machine which features a suite of installed software, including RStudio.

At the end of the course, students were asked to complete a brief survey to provide feedback on the course, including the topics covered, the difficulty of the lessons, course pacing, and RStudio access. Students who took the formal course for which this

workshop was tailored were given a survey following completion of the formal course in May.

Table 1a. Topics covered during the R programming for Economics Majors workshop.

Section	Topics Covered
Using the Interactive R Document	Introduction to RStudio IDE
	Introduction to the learnr format
	R as a calculator
Data types	Types of data in R
	Checking data types with <code>class()</code> , <code>str()</code> , <code>is.<type>()</code>
	Conditional operators
	Converting data types
Variables	Assigning variables with <code><-</code> and <code>=</code>
	<code>print()</code>
Built-in Functions	Introduction to functions
	Getting help with <code>?</code> and <code>??</code>
Vectors	Data types and vectors
	Creating vectors with <code>seq()</code> and <code>c()</code>
	Vector arithmetic
	Naming elements
	Subsetting vectors
	Conditional and pattern selection of vectors
Loops	for loops
	Iterating over elements
	while loops
Matrices	Data types and matrices
	Creating a matrix by combining vectors
	Adding row and column names
	Subsetting matrices
	Matrix arithmetic

Table 1b. Topics covered during the R programming for Economics Majors workshop (cont.).

Section	Topics Covered
Data frames	Introduction to R built-in data frame
	head(), tail(), and str() functions with data frames
	Creating data frames with matrices and vectors
	Adding rows and columns
	Subsetting data frames
	Re-ordering data frames
	Read in external files to data frame
Data visualization	Base plotting functions in R
	Scatter plots and histograms with plot()
	Plotting with ggplot2
	Aesthetics
	Plot titles and axis labels
	Working with multiple layers
	Point labels
	Histograms with ggplot2

3. RESULTS

The course was offered in January 2022, a week before the start of the formal class. 75 students registered for the workshop, and tracked participation ranged from 53 to 73 students throughout the duration of the course. Post-course analyses found that out of the 53 students who participated in every session of the workshop, 44 students (83%) had completed 80% of the hands-on exercises. The students had a preference for working through the Texas A&M R installation that was tied to their student accounts. Most students accessed RStudio through VOAL or on their own personal machines, although a sizable percentage (approximately 20%) used RStudio Cloud. Preparatory materials were distributed prior to the day of the course showing students how to use these interfaces. Students using these interfaces faced some “expected” technical difficulties during the first section of the course where they were first introduced to accessing RStudio and running the ‘learnr’ document. These were mitigated by creating breakout rooms in Zoom where students could get one-on-one help from one of the participating instructors. Most of the initial issues were students having difficulty gaining access to VOAL as well as some difficulties installing RStudio and required packages on personal machines. Such technical issues were common in the formal class taught in the Economics program. As such, in anticipation of initial delays, the first section of the course was designed to be very brief, and the rest of the course ran smoothly once students had managed to run the ‘learnr’ document.

3.1 Description of the Course Content by Section

3.1.1 Introduction

Students were introduced to the RStudio IDE, gaining familiarity with navigating RStudio, how to create new R scripts, how to install packages, and how to view the file system and environment. Students learned how to launch the interactive Shiny document and were introduced to the learnr format. Students were instructed on mathematical operators in R, and exercises in this section included using R as a calculator for simple equations as well as how to call some higher-level math functions (e.g., atan(), sqrt()).

3.1.2 Data Types

In this section, students were introduced to four common data types in R: logical, numeric, integer, and character. The functions str() and class() were introduced in guided exercises, and students used the two functions to check the data types of several elements from a list that contained numerical and character data. The is.logical(), is.numeric(), is.integer(), and is.character() functions were demonstrated, and students completed an independent exercise that required the use of each. The students also conducted independent exercises to evaluate the result of conducting mathematical operations on differing data types. Students were introduced to conditional operators in R and completed independent exercises using the operators to compare numeric, integer, and logical data. Finally, the students were shown the functions as.integer(), as.numeric(), as.logical(), and as.character() to learn how to convert data types, and they completed independent exercises to reinforce the material.

3.1.3 Variables

Students were introduced to variables in R and learned how to assign values to variables. The print() function was introduced in a guided exercise, and students used this function along with class() and str() to inspect variables. Independent exercises in this section provided code with errors that they were asked to fix. The students also learned how to modify the values of variables through mathematical operations and how to assign the value from one variable to another.

3.1.4 Built-in Functions

In this section, students were introduced to the concept of built-in functions in R. The functions sqrt(), atan(), and round() were used as examples. The use of ‘?’ and ‘??’ to get help with functions was covered, and the students used this information to get help with the function head() with which they were asked to examine the built-in iris dataset.

3.1.5 Vectors

Students were introduced to the vector data structure in R and were shown again the different data types that can comprise a vector, reinforcing information that was covered previously in the course. Guided exercises were used to teach the students how to create vectors in r using the ‘:’ operator and seq() function. The students then completed an independent exercise in which they were asked to create and inspect a new vector. We next instructed the students on creating vectors using the c() function and then asked the students to complete an independent exercise that involved creating two new vectors with this method. We then introduced the concept of vector arithmetic with a guided exercise and asked the students to complete additional independent exercises in which they learned how R handles vector arithmetic when a longer vector is not a multiple of a shorter one. Students were next taught how to name the elements of a vector using the names() function in a guided

exercise. We next covered how to select elements of a vector both by slicing and by using the names of the element. The students were then asked to complete an independent exercise in which they created a new vector by combining subsets of two provided vectors. Lastly, the students were instructed on conditional and pattern selection of vectors through several guided exercises and were asked to use both of those in an independent exercise to subset a vector of real estate values based on price and name.

3.1.6 Loops

Students were taught the basic concept of loops and their structure in R. ‘For loops’ were covered first through several guided exercises by either printing or conducting a simple arithmetic operation on a vector of numbers. The students were then asked to complete an independent exercise wherein they first created a ‘for loop’ to print only odd numbers from 1 through 21 and then printed the total number of odd numbers their ‘for loop’ identified. We next asked the students to complete an independent exercise in which an amount of interest accrued within a year with different starting values. We then instructed the students on how to iterate over a vector of characters with a guided exercise and then asked that they complete an independent exercise in which they calculate the property tax for each property within a vector and print out the calculated tax and property name. We next covered the concept of nested ‘for loops’ in a guided exercise and had the students complete an independent exercise in which they created a nested ‘for loop’. Students were then introduced to ‘while loops’ with a guided exercise and were then asked to complete two independent exercises on the topic.

3.1.7 Matrices

In this section, students were introduced to the matrix data structure in R and were taught how to create, modify, and subset matrices. The students first learned to create a matrix with the function `matrix()` and by combining vectors with the `c()` function in a guided exercise. We then covered naming rows and columns using `rownames()` and `colnames()`, and to add additional rows and columns with `cbind()` and `rbind()`. Students were asked to practice this skill by adding a column to an existing matrix in an independent exercise. Next the students learned how to subset a matrix by supplying indices and were asked to practice this technique in an independent exercise. The students were then introduced to the built-in functions `sum()`, `rowSums()`, and `colSums()`, and were asked to practice using these functions in an independent exercise using a provided matrix. In the last portion of this section the students learned how R conducts matrix multiplication. The students were then asked to complete a final independent exercise in which they had to create two new matrices from data provided in a table and then conduct a series of calculations using each matrix.

3.1.8 Data Frames

Data frames in R are in the format of a table or two-dimensional array. They differ from matrices in that they can store heterogeneous data, including numeric, factor, character, and logical types. The data frame is a fundamental data structure in R and is used by many (if not most) R packages to store required data.

The students will likely work with data sets that have a variety of data types in the future. Thus, we started with teaching them some R built-in data frames with different data types and how to do initial assessment with `head()`, `tail()` and `str()` functions. Then, the students learned how to create their own data frames from a matrix with the `as.data.frame()` function and by combining vectors with the `data.frame()` function. They were also taught how to change the

sequence of the columns. Next, `rbind()` and `cbind()` functions were introduced just like we did previously with matrix; these allow one to append a row or column to a data frame. The students also learned to subset and filter portions of a matrix with different methods and how to order a data frame based on the values of one of the columns with the `order()` function. Nowadays, most data lives in data files. Therefore, we taught students how to read the data in a file to a data frame at the end.

3.1.9 Data Visualization

One of the reasons R has gained so many users is that it is a powerful platform for generating high quality graphs and figures from analyzed data. The base plotting system in R, however, is somewhat limited. We briefly covered the base functions for plotting in R, and the students learned how to generate a scatter plot and add titles and axis labels in a guided exercise using data from the built-in `mpg` dataset. This was reinforced by an independent exercise in which the students generated plots using different data from the ‘mpg’ data frame. Given the somewhat limited nature of the R base plotting functions, we chose to teach plotting with the popular R package ‘`ggplot2`’. The students learned the basic structure of the code for plotting with ‘`ggplot2`’ in a guided exercise, wherein we generated another scatter plot using data from the ‘mpg’ data frame. We then used these results to discuss how ‘`ggplot2`’ plots in layers and to discuss the concept of aesthetics in ‘`ggplot2`’. We then asked the students to complete an independent exercise that entailed using ‘`ggplot2`’ to create another scatter plot on their own. Following this, the students learned how to change point shapes, fill, size, adding plot titles and axis labels, and adding aesthetics mapped to different variables in a series of guided and independent exercises. We next covered how to add additional layers to a plot and finally how to add point labels using `geom_text()` and `geom_label()` through guided exercises. We finished this section by teaching the students how to generate histograms using ‘`ggplot2`’.

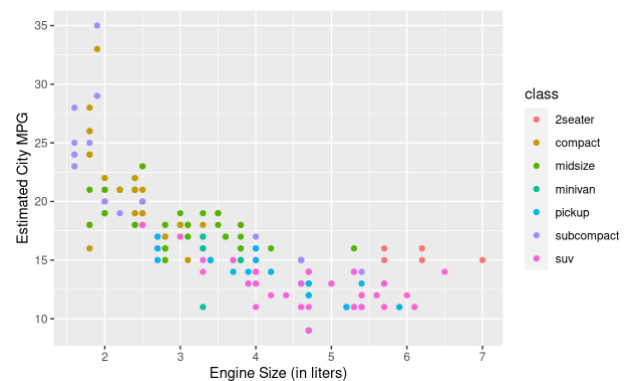


Figure 1. Scatter plot generated in an independent exercise during the Data Visualization section of the R programming for Economics Majors course.

3.2 Feedback from Students

A survey conducted following the end of the course revealed predominantly positive feedback. A vast majority of students (94.3%) felt that the course covered a sufficient range of topics, and most students (87.4%) found that the exercises were at the appropriate level of difficulty. A majority of the students reported that they were engaged for each topic, although many reported needing additional time to complete the exercises as the course progressed through more difficult material (Figure 2). Over half

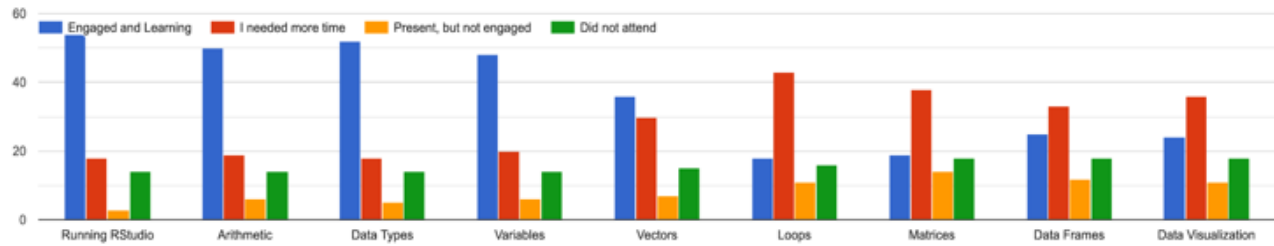


Figure 2. Student self-reported levels of engagement of each topic covered throughout the informal short course.

(51.5%) of the students feel they would have benefited from an additional day of training.

We taught the students different methods of running RStudio (as shown in Figure 3), which gave them more flexibility to choose a favorable platform for the formal economic courses. Most students used either the VOAL option for running RStudio or ran RStudio on their own personal machine. The VOAL option was beneficial, as RStudio was already installed, and installing the packages necessary to run the learnr document was relatively easy for the students, as most of the dependencies for this package were already installed on the VOAL machines. However, interactive feedback during the course as well as survey results revealed that interacting with RStudio through VOAL was frustratingly slow. Given that most students use their personal machines for running RStudio during formal economics courses, it may prove beneficial to direct students to use this approach in future offerings of this workshop. This might be achieved by having a session before the workshop during which students can receive help with installing RStudio and the packages necessary for the workshop.

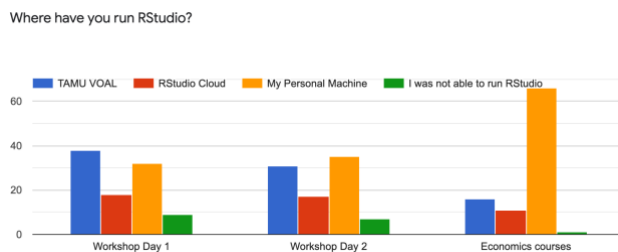


Figure 3. Student responses to where they ran RStudio during the course and where they ran RStudio during formal courses.

The survey following completion of the formal economics course offered valuable insight into changes that should be addressed in future iterations of this workshop. Particularly, one open-ended question asked, “In retrospect, what would you change in the pre-course two-day R workshop for it to be more useful in terms of your performance in this course?” Several key points were common in the students’ responses. First, many complained about technical difficulties that were encountered during the workshop. These included the issues previously discussed with the connection to VOAL and problems with installation on personal machines. This being one of the most frequent comments following a semester of formal coursework highlights the importance in addressing it in future offerings of this course. Another frequent response on the survey regarded the length of the workshop. Many students commented that staying engaged in a virtual workshop for 12 hours spread across two days was very difficult, and many felt that they would have benefitted from additional contact hours. This problem can easily be solved given the modular nature of the design of the course. The course can easily be adjusted to be taught for fewer

hours per day across more days. Also, moving the course from a virtual setting to an in-person one will further improve engagement. Lastly, another common request from the students was to include more instruction on economics-related functions and code. This course was designed to give the students a strong foundation in R prior to a formal course in which R was utilized. In fact, many students commented that they were entering with no knowledge of R. In order to address the needs of students with little or no knowledge of R, the existing modules for the course should be maintained (and perhaps expanded). However, if the course was expanded into more contact hours, it might be worth adding additional modules that include R packages the students might use during the formal course. This might be limited, however, by the need for discipline-specific knowledge for utilizing these packages. For example, some students requested additional training on coding for economic forecasting, which necessitates the need for knowledge on this topic that is outside the scope of our instructors.

4. CONCLUSION

The work is in continuation of our previous effort to build a continuum between cybertraining and domain-specific courses. Working with the faculty of record, we developed a course during which students were instructed in using the R programming language assuming little or no prior knowledge or experience. The topics covered were tailored for these students based on discussions between members of the Texas A&M High Performance Research Computing group and the Department of Economics. The collaborative efforts of our two groups ensured a strong foundational knowledge of R that could be expanded in following discipline-specific graduate level courses. The course was designed using the ‘learnr’ package in R, which generated a highly portable and easily scalable framework that can be easily adopted and modified for any user group. Students completing the hands-on projects were more likely to complete the course. Feedback from students suggests that this course may best be implemented as several in-person sessions spread across several days. The modular nature of the course makes this implementation easy to achieve and also lends itself to augmentation with additional modules that might cover discipline-specific R packages. In light of the user feedback and growing demand for R training nationwide, the course is being developed as an asynchronous learning resource. Users completing the asynchronous course will be offered the opportunity to earn a micro-credential from TEES-Edge (affiliated with Texas A&M University). The HPRC training program continues to pursue models that leverage campus cyberinfrastructure to positively impact academic curricula.

5. SUPPORTING INFORMATION

The training materials used in this study are available to the community via the Texas A&M HPRC website [13]. The slides used to introduce topics and the R code to render the learnr Shiny interactive document are included in the material. Please send us

feedback on your experience adopting this material in an email to help@hprc.tamu.edu.

6. ACKNOWLEDGEMENTS

This work was supported by the Department of Economics at Texas A&M University; the “Accelerating Microcredentials Program” at the Texas Higher Education Coordinating Board; the National Science Foundation (NSF) award number 1925764, “CC* Cyberteam SWEETER”; NSF award number 2019129, “MRI:FASTER”; NSF award number 1730695, CyberTraining: CIP: CiSE-ProS: Cyberinfrastructure Security Education for Professionals and Students”; NSF award number 1818253, Frontera: Computing for the Endless Frontier; and NSF award number 2019136, “CC* BRICCs: Building Research Innovation at Community Colleges.” We thank Dr. Gan Li, Dr. Lisa Perez, and Ridham Patiyyola for useful discussions.

7. REFERENCES

- [1] Garrick Aden-Buie, Barret Schloerke, and JJ Allaire. 2022. Interactive Tutorials for R • learnr. Retrieved from <https://rstudio.github.io/learnr/>
- [2] Stephen Cass. 2019. The Top Programming Languages of 2019 - IEEE Spectrum. (Sep. 2019). Retrieved from <https://spectrum.ieee.org/the-top-programming-languages-2019>
- [3] Dhruva K. Chakravorty, Marinus “Maikel” Pennings, Honggao Liu, Xien Thomas, Dylan Rodriguez, and Lisa M. Perez. 2020. Incorporating Complexity in Computing Camps for High School Students - A Report on the Summer Computing Camp at Texas A&M University. *Journal of Computational Science Education* 11, 1 (Jan. 2020), 12–20. DOI: <https://doi.org/10.22369/issn.2153-4136/11/1/3>
- [4] Dhruva K. Chakravorty, Marinus “Maikel” Pennings, Honggao Liu, Zengyu “Sheldon” Wei, Dylan M. Rodriguez, Levi T. Jordan, Donald “Rick” McMullen, Noushin Ghaffari, and Shaina D. Le. 2019. Effectively Extending Computational Training Using Informal Means at Larger Institutions. *Journal of Computational Science Education* 10, 1 (Jan. 2019), 40–47. DOI: <https://doi.org/10.22369/issn.2153-4136/10/1/7>
- [5] Dhruva K. Chakravorty, Marinus “Maikel” Pennings, Honggao Liu, Zengyu “Sheldon” Wei, Dylan M. Rodriguez, Levi T. Jordan, Donald “Rick” McMullen, Noushin Ghaffari, Shaina D. Le, Derek Rodriguez, Crystal Buchanan, and Nathan Gober. 2019. Evaluating Active Learning Approaches for Teaching Intermediate Programming at an Early Undergraduate Level. *Journal of Computational Science Education* 10, 1, (Jan. 2019), 61–66. DOI: <https://doi.org/10.22369/issn.2153-4136/10/1/10>
- [6] Dhruva K. Chakravorty, Lisa M. Perez, Honggao Liu, Braden Yosko, Keith Jackson, Dylan Rodriguez, Stuti H. Trivedi, Levi Jordan, and Shaina Le. 2021. Exploring Remote Learning Methods for User Training in Research Computing. *Journal of Computational Science Education* 12, 2 (Feb. 2021), 11–17. DOI: <https://doi.org/10.22369/issn.2153-4136/12/2/2>
- [7] Dhruva Chakravorty and Minh Tri Pham. 2020. Evaluating the Effectiveness of an Online Learning Platform in Transitioning Users from a High Performance Computing to a Commercial Cloud Computing Environment. *Journal of Computational Science Education* 11, 1 (Jan. 2020), 93–99. DOI: <https://doi.org/10.22369/issn.2153-4136/11/1/15>
- [8] Zhenhua He, Jian Tao, Lisa M. Perez, and Dhruva K. Chakravorty. 2022. Technology Laboratories: Facilitating Instruction for Cyberinfrastructure Infused Data Sciences. *Journal of Computational Science Education* 13, 1 (Apr. 2022), 44–49. DOI: <https://doi.org/10.22369/issn.2153-4136/13/1/8>
- [9] Richard Lawrence, Zhenhua He, Wesley Brashear, Ridham Patoliya, Honggao Liu, and Dhruva K. Chakravorty. 2022. Tailored Computing Instruction for Economics Majors. *Journal of Computational Science Education* 13, 1 (Apr. 2022), 32–37. DOI: <https://doi.org/10.22369/issn.2153-4136/13/1/6>
- [10] Richard Lawrence, Tri M. Pham, Phi T. Au, Xin Yang, Kyle Hsu, Stuti H. Trivedi, Lisa M. Perez, and Dhruva K. Chakravorty. 2022. Expanding Interactive Computing to Facilitate Informal Instruction in Research Computing. *Journal of Computational Science Education* 13, 1 (Apr. 2022), 50–54. DOI: <https://doi.org/10.22369/issn.2153-4136/13/1/9>
- [11] Catherine L. Muller and Chris Kidd. 2014. Debugging geographers: teaching programming to non-computer scientists. *Journal of Geography in Higher Education* 38, 2 (Apr. 2014), 175–192. DOI: <https://doi.org/10.1080/03098265.2014.908275>
- [12] Mahwish Shahid, Afifa Wajid, Kamran Ul Haq, Imran Saleem, and Abdul Haseeb Shujja. 2019. A Review of Gamification for Learning Programming Fundamental. In *2019 International Conference on Innovative Computing (ICIC)*, November 1–2, 2019, Lahore, Pakistan, IEEE, 1–8, DOI: <https://doi.org/10.1109/ICIC48496.2019.8966685>
- [13] Texas A&M High Performance Research Computing. R for Economics Graduate Students. Retrieved from <https://hprc.tamu.edu/events/workshops/2022-01-11-REcon.html>